UNIVERSITÉ DE MONTRÉAL

**CENTRE DE RECHERCHE SUR LES TRANSPORTS**

CASE POSTALE 6128, STATION A, MONTRÉAL, QUÉBEC, H3C 3J7

# EFFICIENT IMPLEMENTATION OF HEURISTIC ALGORITHMS FOR THE QUADRATIC ASSIGNMENT PROBLEM

by

Marc Los

An earlier version of this paper was presented at the International Symposium on Locational decisions, at Banff, Alberta, Canada, on April 24th-28th, 1978

Université de Montréal

Publication # 87 - Centre de recherche sur les transports
Cahier # 7712 - Département de sciences économiques

Décembre 1977 - Révisé en février 1979

Dépôt légal: 2e trimestre 1979

## ABSTRACT

This paper presents computational results obtained with an efficient new implementation of the steepest descent pairwise interchange technique for the quadratic assignment problem. This technique is compared with two other heuristic techniques. The computational efficiency of this new code is due to the use of a recursive technique which appears empirically to cut down time complexity from $O(n^4)$ to $O(n^3)$, (where $n$ is the size of the problem).

## I Introduction

The quadratic assignment problem can be used in many fields, such as plant layout [9] or urban planning [3,13,17,18]. This problem was first mentioned by Koopmans and Beckmann [14] and generalized by Lawler [15]. The latter author proposed a branch and bound technique to solve that problem [15] and other authors have proposed different approaches [6,7,10,20]. So far no exact technique has been able to handle problems with more than 16 activities and it is doubtful whether improved exact techniques could handle problems of a realistic size which might include as many as 70 or 80 activities. Practical algorithms therefore have to be of a heuristic nature.

Two types of heuristic techniques have been proposed: constructive techniques [11,16] or hill-climbing techniques, usually coupled with random generation of an initial spatial allocation [1,4,12]. Several comparisons have been made of these different heuristic techniques and have previously been reported [5,19,22]. Also in [9], Francis and White give an exhaustive presentation and comparison of existing algorithms, exact or heuristic. These reviews and comparisons indicate clearly that hill-climbing techniques produce solutions of a much better quality than constructive techniques. In particular, the so-called steepest-descent pairwise interchange procedure is considered to be the best heuristic technique among the hill-climbing algorithms proposed to date. However it is usually stated that this technique requires too much computational effort to be applied to realistic problems [1].

---

1. See [9] Chapter 8 for a discussion of that point of view.

This paper will show that this is not necessarily so if one uses an efficient recursive technique to cut down the time-complexity of the steepest-descent pairwise interchange procedure from $O(n^4)$ to $O(n^3)$ where n is the size of the problem. This new code will then be compared to a steepest-descent 3-way interchange procedure (which can be considered as a generalization of the pairwise exchange procedure)and to a constructive technique based on results by Graves and Whinston [11]. The comparison between the pairwise interchange procedure and the "Graves-Whinston" constructive technique is purely illustrative since there may be better constructive techniques. No attempt however will be made at an exhaustive comparison of all existing constructive techniques with the proposed hill-climbing technique, given that such comparisons have already been made by other authors.

The rest of the paper is organized as follows. Section II states the quadratic assignment problem, presents the pairwise and three-way interchange procedures as well as the constructive technique. Recursive techniques are used to speed up the computational efficiency of these three algorithms. Section III presents the results of the numerical experiments and section IV concludes the paper.

## II The quadratic assignment problem and proposed solution algorithms

### Mathematical formulation

The quadratic assignment problem will be stated in its Koopmans-Beckmann formulation, using notation previously employed by Graves and Whinston [11] The matrices will be given meanings in terms of a land use planning problem. We have n activities (e.g. residential units, employment centers, shopping centers) to assign to n discrete locations, so as to minimize a total cost

composed of site costs and interactions costs. The following matrices are given exogenously:

$A = [a_{ij}]$ nxn matrix

$a_{ij}$ = site cost of assigning activity i to location j (e.g. a development cost)

$Q = [q_{ij}]$ nxn matrix

$q_{ij}$ = interaction between the activities i and j (e.g. the number of trips between the activities i and j per time unit).

$C = [c_{ij}]$ nxn matrix

$c_{ij}$ = cost per interaction unit between the locations i and j (e.g. shortest path cost over a transportation network)

There are n! possible one-to-one mappings $\rho$ of the set of activities into the set of locations. Let $\rho(i)$ be the location assigned to the activity i in the mapping $\rho$ and let $\delta(k) = \rho^{-1}(k)$ be the activity assigned to location k in the inverse mapping $\delta = \rho^{-1}$. To each mapping $\rho$ is associated a cost $\phi(\rho)$:

$$\phi(\rho) = \sum_{i=1}^{n} a_{i\rho(i)} + \sum_{i=1}^{n} \sum_{j=1}^{n} q_{ij} c_{\rho(i)\rho(j)} \qquad (1)$$

The first summation term represents the site costs, while the second summation term represents the interaction costs. The problem is to find an assignemnt $\rho^*$ which minimizes $\phi(\rho)$.

The steepest-descent pairwise interchange procedure

The idea of this method is due to Armour and Buffa [1] and was implemented originally in a program called CRAFT. The method works as follows:

a) start with an initial spatial allocation (obtained by generating a

random permutation for instance);

b) among all possible $\frac{n(n-1)}{2}$ pairwise interchanges of activities, choose for implementation that interchange which brings the greatest decrease to the objective function. Stop if no pairwise interchange of activities brings an improvement. Otherwise go to step c.

c) go back to the beginning of step b.

In step b it is necessary to evaluate $\frac{n(n-1)}{2}$ potential cost changes, and it is necessary to evaluate such cost changes each time a pairwise interchange of activities has been implemented. It is this evaluation that can be time-consuming, as mentioned by Francis and White ([9], chapter 8). However, when at least one interchange has been implemented, it is possible to update the matrix of cost changes, by making use of the previous matrix, in a recursive manner, thus making the subsequent iterations less time-consuming than the first. The method can be described as follows[1]. We define

$$w_{ij} = q_{ij} + q_{ji} \tag{2}$$

$$b_m^{ij} = c_{im} - c_{jm} \tag{3}$$

Let $C_{\ell m}$ be the cost (or benefit) of interchanging the activities currently located at $\ell$ and m, taking only interaction costs into account. The subscripts $\delta(k),\delta(m)\ldots$ refer to the spatial allocation before implementation of the interchange considered

$$C_{\ell m} = \sum_{k=1}^{n} b_k^{\ell m} \times [w_{\delta(k)\delta(m)} - w_{\delta(k)\delta(\ell)}] + 2 \times c_{\ell m} \times w_{\delta(\ell)\delta(m)} \tag{4}$$

Suppose that one interchange has actually occurred between the activities located at i and j. Let $\Delta_{ij}[C_{\ell m}]$ be the change in $C_{\ell m}$ after this inter-

1. The cost matrix is assumed symmetrical, but the same method can be applied in the case of an asymmetrical matrix.

change. One can derive the following expression for $\Delta_{ij}[C_{\ell m}]$:

$$\Delta_{ij}[C_{\ell m}] = [b_{\ell}^{ji} + b_{m}^{ij}] \times [w_{\delta(i)\delta(m)} + w_{\delta(j)\delta(\ell)} - w_{\delta(j)\delta(m)} - w_{\delta(i)\delta(\ell)}] . \quad (5)$$

for all $\ell$ such that $\ell \neq i$ and $\ell \neq j$, for all m such that $m \neq i$ and $m \neq j$. Therefore, after an interchange, the new value of $C_{\ell m}$, called $C'_{\ell m}$, is:

- When $\ell = i$ and $m = j$:

$$C'_{\ell m} = -C_{\ell m} \quad (6)$$

- When $\ell = i$ or $\ell = j$, or when $m = i$ or $m = j$, $C'_{\ell m}$ is computed by expression (4)

- When $\ell \neq i$ and $\ell \neq j$, $m \neq i$ and $m \neq j$.

$$C'_{\ell m} = C_{\ell m} + \Delta_{ij}[C_{\ell m}] \quad (7)$$

where $\Delta_{ij}[C_{\ell m}]$ is given by (5).

At each iteration of the pairwise interchange technique we have to compute $n(n-1)/2$ expressions $C'_{\ell m}$ (The updating of the site costs is trivial). Therefore the first iteration which necessarily uses (4) has the time complexity $O(n^3)$, while each subsequent iteration, which uses (7) (with few exceptions) has the time complexity $O(n^2)$. The steepest-descent pairwise interchange algorithm was implemented in a FORTRAN program called ACTIV.

The steepest-descent three-way interchange procedure

This algorithm generalizes the pairwise interchange technique by taking into account three-way permutations of the activities, in addition to pairwise interchanges.

The key to an efficient implementation of that algorithm is found again in a recursive mechanism to update the potential cost-changes at step b.[1] This technique was implemented in a Fortran program called SPAL.

## A constructive technique

A constructive technique was designed and implemented for purposes of comparison with the two hill-climbing techniques described previously. It assigns activities to locations sequentially and is based on results by Graves and Whinston [11].

At the first iteration the first activity is assigned to each potential location in turn and for each location the mean of the objective function over all possible remaining assignments is computed. The first activity is eventually assigned to the location corresponding to the smallest mean. More generally, at the $k^{th}$ iteration, the $k^{th}$ activity is assigned to each unassigned potential location in turn, the mean of the cost over all remaining possible assignments is computed and the assignment with the smallest associated mean is chosen for activity k. The algorithm stops when all activities have been assigned to all locations.

---

1. Although more complex, the recursive updating mechanism of the cost changes is analogous to the pairwise interchange method and will not be described in detail here (see 17).

The derivation of the mean is found in [11]. The mean is computed in a recursive way at each iteration [17]. This technique was implemented in a Fortran program called GRAWHI.

## III Numerical experiments

The numerical experiments aimed at determining the efficiency of ACTIV as well as the quality of the solutions obtained by that method. ACTIV was also compared with SPAL and GRAWHI. A combined use of ACTIV and SPAL was also attempted. For some problems of small size the global optimum was computed using a code developed by the present author for a branch and bound technique proposed by Lawler [15]. This allowed the computation of the percentage deviation between the value of the heuristic solution and the value of the exact optimum in certain cases.

Four series of data were used for testing the algorithms: firstly, a series of test problems used by the author in a previous work [17]. (This first series will be called A in the rest of this paper); secondly, the Steinberg problem [23] also used by Graves and Whinston [11]; thirdly, two series of randomly generated problems (called series B and series C in the rest of the paper). For problems of series B or series C, n points were generated in a square of side 100 units of length, using a uniform distribution. The Euclidean pairwise distances were computed and a network was built containing only a subset of the links of the complete network (of $\frac{n(n-1)}{2}$ links). For each node in turn only the links linking that node to the closest nodes were retained so as to have 6 links per node in the end. The interaction costs $c_{k\ell}$ were then computed as the shortest path lengths in the

network. Each site cost $a_{ik}$, and each interaction $q_{ij}$ were generated from uniform distributions such that:

$$0 \leq a_{ik} \leq 100 \quad \text{and} \quad 0 \leq w_{ij} = q_{ij} + q_{ji} \leq 10\,000.$$

Efficiency of the algorithm

For different values of n, $(10 \leq n \leq 72)$, several problems were generated, and for each problem several permutations were randomly generated and used as initial solutions for the pairwise interchange technique. Thus several local optima were generated for each problem using ACTIV. The results of these experiments are summarized in Table 1, Figure 1 and Figure 2. Several observations can be made.

- There appears to be a linear relationship between the number of pairwise interchanges implemented during the algorithm and the size n of the problem.
- Taking into account this empirical result, and the results of section II the time complexity of ACTIV appears to be $O(n^3)$.
- The execution time for a problem of size $n = 72$ was only 89.32 seconds of CDC Cyber 173.

  The storage space utilized by ACTIV is $(8n^2 + 4n)$ words.

A second series of problems were generated (series C) to which ACTIV, SPAL and GRAWHI were applied. The results of these experiments are summarized in Table 2, Figure 3 and Figure 4.

Several observations can be made:

- There appears to be a linear relationship between the number of exchanges implemented and the size of the problem, for SPAL as well as for ACTIV. For SPAL, however, less exchanges were necessary to reach a local optimum than for ACTIV.
- ACTIV is more efficient than GRAWHI and much more efficient than SPAL, in terms of execution time. For instance, ACTIV takes 2.12 seconds of CDC Cyber 173 for a problem of size n = 24, while GRAWHI takes 5.45 seconds and SPAL 27.70 seconds. The same remark can be made concerning storage space, since GRAWHI utilizes $9n^2 + 4n$ words and SPAL $[4n^3 + 8n^2 + 4n + \frac{1}{3}n(n-1)(n-2)]$ words, to be compared with $(8n^2 + 4n)$ for ACTIV.

Quality of the solutions obtained

For the problem of size n=10 (series C), the global optimum was computed. GRAWHI obtained the optimal solution only once: on the average the percentage deviation from the optimum was 5.38%. ACTIV obtained the optimal solution for 15 of the 20 problems: on the average the percentage deviation from the optimum was .062% (Table 3). SPAL obtained the global optimum for 18 of the 20 problems: on the average, the percentage deviation from the optimum was .006% (Table 3). Only for one problem did ACTIV produce a better local optimum than SPAL. These findings carried over to problems of greater size. GRAWHI always produced much worse results than either SPAL or ACTIV and therefore the experiments

with GRAWHI are not reported in detail here. Tables 2, 3 and 4 clearly indicate that SPAL tends to give solutions of only a slightly better quality than ACTIV. These results are confirmed by experiments on series A problems (Tables 5 and 6). Therefore one can conclude that GRAWHI can be eliminated because it is not even better than ACTIV on efficiency grounds. SPAL and ACTIV are comparable for solution quality though SPAL tends to give slightly better solutions. However SPAL i' much more time-consuming and uses storage space much more than ACTIV and is therefore not suitable for large size problems ($n > 24$).

Combined use of ACTIV and SPAL

For problems of small size ($n \leq 24$) one can try to combine ACTIV and SPAL in order to obtain better solutions than either algorithm alone could obtain. This was tried. For each of the 5 problems (# 4, 10, 13, 14, 18) of size $n = 10$ (series C) for which ACTIV could not reach the global optimum, the best locally optimal solution given by ACTIV was used as an initial solution for SPAL. Only for problem # 13, did SPAL produce an improvement: SPAL produced the global optimum and it took .408 seconds and 3 three-way exchanges. For problem # 4, of size $n = 16$ (series C) SPAL was also applied using the best local optimum given by ACTIV as an initial solution: no improvement occurred. For 2 problems of size $n = 24$ (series C) SPAL was applied, using the best local optimum produced by ACTIV as an initial solution, in each case. No improvement occurred for problem # 1. However for problem # 2, an improvement occurred and a solution was obtained that was better than those produced by either ACTIV or SPAL alone: The results are summarized in Table 7. It can

be checked that SPAL reaches the new local optimum in only three exchanges. Thus a better solution could be reached by (ACTIV + SPAL) in 10.7 seconds, than by SPAL alone in 35.498 seconds. These results tend to justify the use of SPAL as a complement to ACTIV rather than as an alternative to ACTIV.

Results obtained by ACTIV on the Steinberg problem

Table 8 reports the results produced by ACTIV for the Steinberg problem [11,23] The best locally optimal solution obtained is given below.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 33 | 32 | 19 | 14 | 15 | 16 | 1 | 9 | 35 |
| 34 | 30 | 28 | 20 | 11 | 7 | 10 | 18 | 17 |
| 31 | 29 | 27 | 23 | 12 | 13 | 4 | 8 | 2 |
| 22 | 21 | 24 | 25 | 26 | 6 | 5 | 3 | 36 |

Structure of the best locally optimal solution generated by ACTIV (of value 4211.40)

Each run takes less than 2 seconds of CDC CYBER 74 and improves the initial solution by approximately 50% in all cases. Among the fifteen local optima obtained, four are better than the solution reported by Graves and Whinston [11]. Besides, ACTIV was also applied to the Steinberg problem, using as initial solution the solution reported by Graves and Whinston. The results are reported below, and one can see that it is possible to improve Graves and Whinston's solution.

| Value of the Graves and Whinston assignment | Value of the local optimum given by ACTIV | Execution time in seconds (CDC CYBER 74) | Number of exchanges implemented |
|---|---|---|---|
| 4344.975 | 4236.497 | 0.344 | 3 |

Exchanges implemented:
Locations of activities exchanged

22 and 12, 29 and 20, 17 and 8

Results of ACTIV on the Steinberg problem, with the solution given by Graves and Whinston as an initial solution

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 36 | 16 | 15 | 29 | 28 | 19 | 30 | 34 | 31 |
| 35 | 3 | 7 | 13 | 12 | 14 | 32 | 33 | 21 |
| 2 | 9 | 10 | 1 | 11 | 20 | 23 | 25 | 22 |
| 17 | 18 | 8 | 4 | 5 | 6 | 27 | 26 | 24 |

Structure of the locally optimal solution given by ACTIV with Graves and Whinston's assignment as an initial solution

## IV Conclusions

This paper presented an improved version of the steepest descent pairwise interchange algorithm for the quadratic assignment problem. For n activities, the time complexity of this technique appears to be $O(n^3)$ instead of $O(n^4)$ with previous versions. This efficiency is due to a recursive method of updating the $n(n-1)/2$ changes in the total cost that have to be evaluated at each iteration of the algorithm. It is reflected in the fact that it took less than 2 seconds of CDC CYBER 74 to find a local optimum to the Steinberg problem ($n = 36$) and 89.32 seconds of CDC CYBER 173 to find a local optimum to a problem of size $n = 72$. Besides, the comparison of the solutions obtained by this technique with the optimal solutions, for small problems of size $n \leq 12$ for which the optimal solutions were known, corroborated the finding of previous authors [5,9] that solutions of very good quality are obtained by this method. This was confirmed by the results obtained with the Steinberg problem, for which a better solution was obtained than had been reported before.

The pairwise interchange technique was generalized to include three-way permutations of activities. A recursive technique was also used to speed up computation. In spite of that, experimental results show that the method is not applicable to problems of size $n > 24$, because of the time and storage requirements. In addition, this technique does not improve by much the quality of the solutions that can be obtained by the pairwise interchange technique alone. However it can be used in combination with the latter to find better solutions than could be obtained by either method alone. This combined technique is applicable only for small problems ($n \leq 24$).

To summarize, one should use a pairwise interchange technique such as reported in this paper to solve problems of a realistic size ($n \geq 24$) Several local optima should be obtained by generating at random the initial permutations. For small problems ($n \leq 24$) a three-way interchange technique should be used to try to improve the best local optimum obtained by the first technique.

The method proposed in this paper can be generalized to include asymmetric cost matrices (such matrices can occur in the case of air pollution costs) as well as several types of interactions (transportation interactions, air pollution interactions, etc.). In the latter case the time complexity is a linear function of the number of interactions: the same recursive method can be used to update separately cost changes of each type. Indeed such a method was implemented and used to solve large size land use planning problems [13]. [1]

---

1. The codes mentioned in this paper are available upon request to the author.

## REFERENCES

[1] Armour, G. and Buffa, E.: "A Heuristic Algorithm and Simulation Approach to Relative Location of Facilities". Management Science 9: 294-309 (1963).

[2] Bazaraa, M.S.: "The Use of Bender's Partitioning Scheme for Solving the Quadratic Assignment Problem". Presentation at the San Francisco Joint ORSA/TIMS Conference, May 1977.

[3] Brotchie, I.F., Sharpe, R.: "Hierarchical Model for Urban and Regional Planning", in "Urban and Regional Models". Proceedings of the PTRC Seminar on Urban and Regional Models, University of Warwick, U.K., 10-12 July 1974.

[4] Buffa, E.S., Armour, G.C. and Volmann, T.E.: "Allocating Facilities with CRAFT", Harvard Business Review 42 (March/April 1964).

[5] Burkard, R.E. and Stratmann, K.H.: "Numerical Investigations on Quadratic Assignment Problems". N.R.L.Q., 1978, Vol. 25, No. 1, pp. 129-148.

[6] Christofides, N. and Gerrard, M.: "A Graph Theoretic Analysis of Bounds for the Quadratic Assignment Problem". January 1978, Imperial College, London, England.

[7] Christofides, N. and Gerrard, M.: Special Cases of the Quadratic Assignment Problem", 1978, Imperial College, London, England.

[8] Francis, R.L. and Goldstein, J.M.: "Location Theory: A Selective Bibliography", Operations Research, 22:2, 400-410 (1974).

[9] Francis, R.L. and White, J.A.: "Facility Layout and Location - An Analytical Approach" Prentice-Hall, 1974.

[10] Gilmore, P.C.: "Optimal and Suboptimal Algorithms for the Quadratic Assignment Problem" J. SIAM Vol. 10, pp. 305-313 (1962).

[11] Graves, G. and Whinston, A.: "An Algorithm for the Quadratic Assignment Problem", Management Science, 17, # 7: 453-77 (March 1970).

[12] Hillier, F.S. and Connors, M.M.: "Quadratic Assignment Problem Algorithms and the Location of Indivisible Facilities", Management Science 13, # 1: 42-57 (1967).

[13] Hopkins, L.D.: "Land-Use Plan Design - Quadratic Assignment and Central Facility Models". Environment and Planning A, Vol. 9, pp. 625-642 (1977).

[14] Koopmans, T. and Beckmann, M.: "Assignment Problems and the Location of Economic Activities" Econometrica 25, # 1: 53-76 (January 1957).

[15] Lawler, E.C.: "The Quadratic Assignment Problem", Management Science 9, 586-99 (1963).

[16] Lee, R.C. and Moore, J.M.: "CORELAP - Computerized Relationship Layout Planning". Journal of Industrial Engineering, Vol. 18 # 3, 195-200 (1967).

[17] Los, M.: "Simultaneous Optimization of Land Use and Transportation in New Town Design". Ph.D. Dissertation in City and Regional Planning, University of Pennsylvania, May 1975.

[18] Los, M.: "Simultaneous Optimization of Land Use and Transportation: A Synthesis of the Quadratic Assignment Problem and of the Optimal New-work Problem" (1977). Regional Science and Urban Economics 8, 1978, pp. 21-42.

[19] Nugent, C.E., Volmann, T.E. and Ruml, J.: "An Experimental Comparison of Techniques for the Assignment of Facilities to Locations". Operations Research 16, # 1 (January/February 1968).

[20] Picard, J.C. and Queyranne, M.: "Exact Algorithms for the Quadratic Assignment Problem". Département de génie industriel, Ecole Polytechnique de Montréal, August 1976.

[21] Pierce, J.F. and Crowston, W.B.: "Tree-Search Algorithms for the Quadratic Assignment Problem". N.R.L.Q., Vol. 18, pp. 1-36 (1971).

[22] Ritzman, L.: "The Efficiency of Computer Algorithms for Plant Layout". Management Science 18, # 5, (January 1972).

[23] Steinberg, L.: "The Backboard Wiring Problem: A Placement Algorithm". SIAM Review 3, # 1 (January 1961).

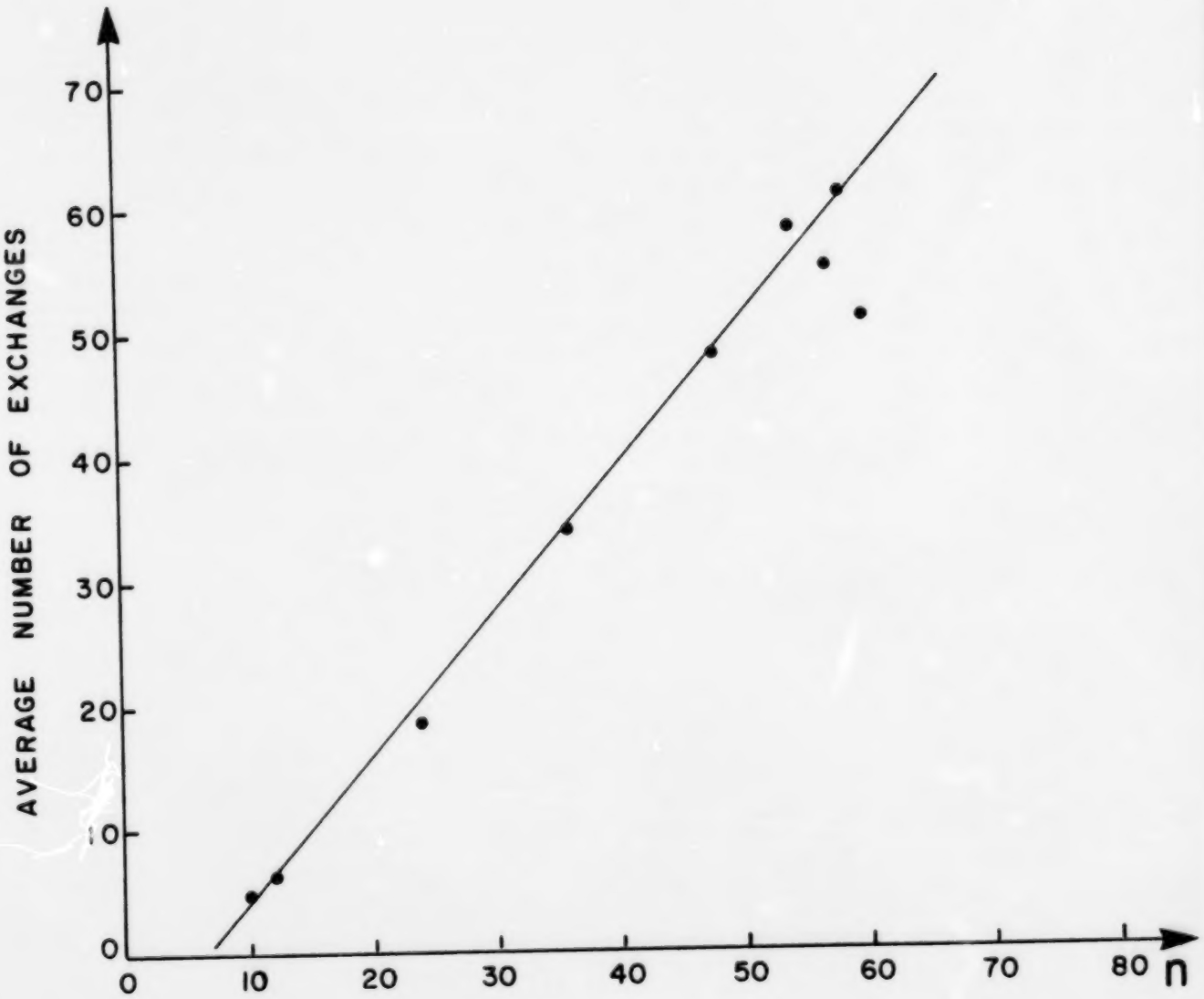


Figure 1: Average number of exchanges implemented by ACTIV as a function of the size of the problem (n)

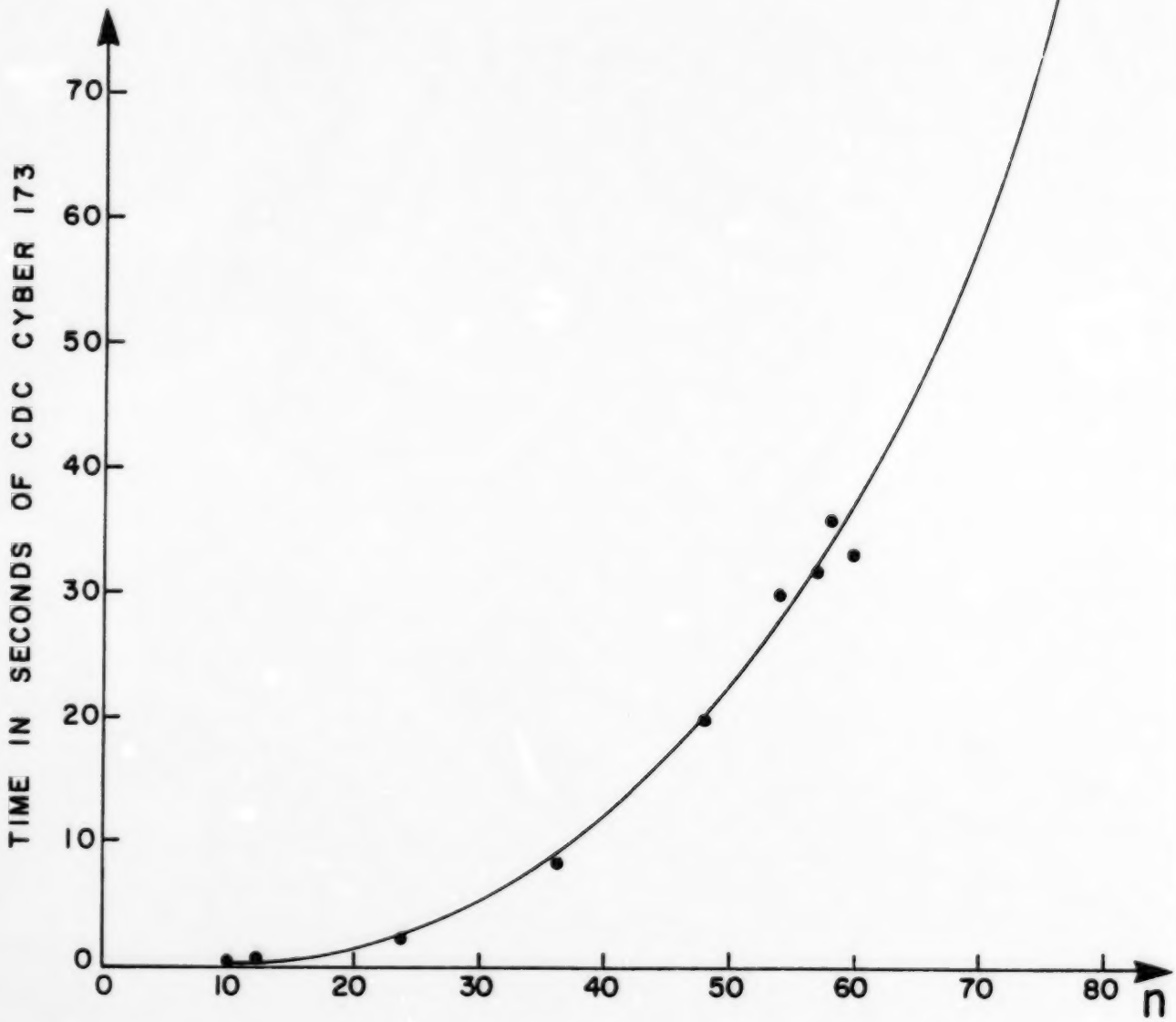


Figure 2: Execution time of ACTIV as a function of the size of the problem (n)

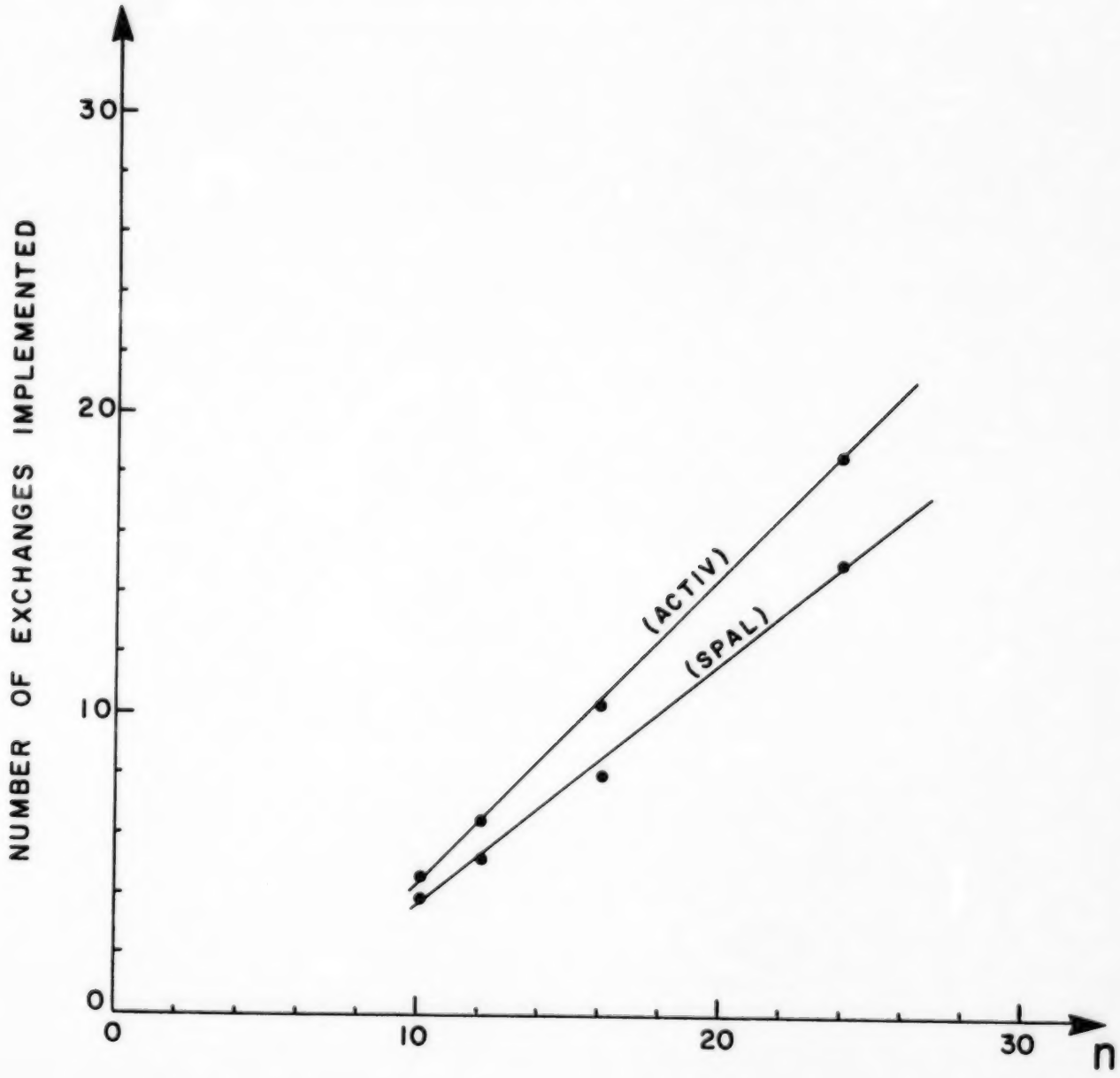


Figure 3: Relationship between the number of exchanges and the size of the problem for ACTIV and SPAL

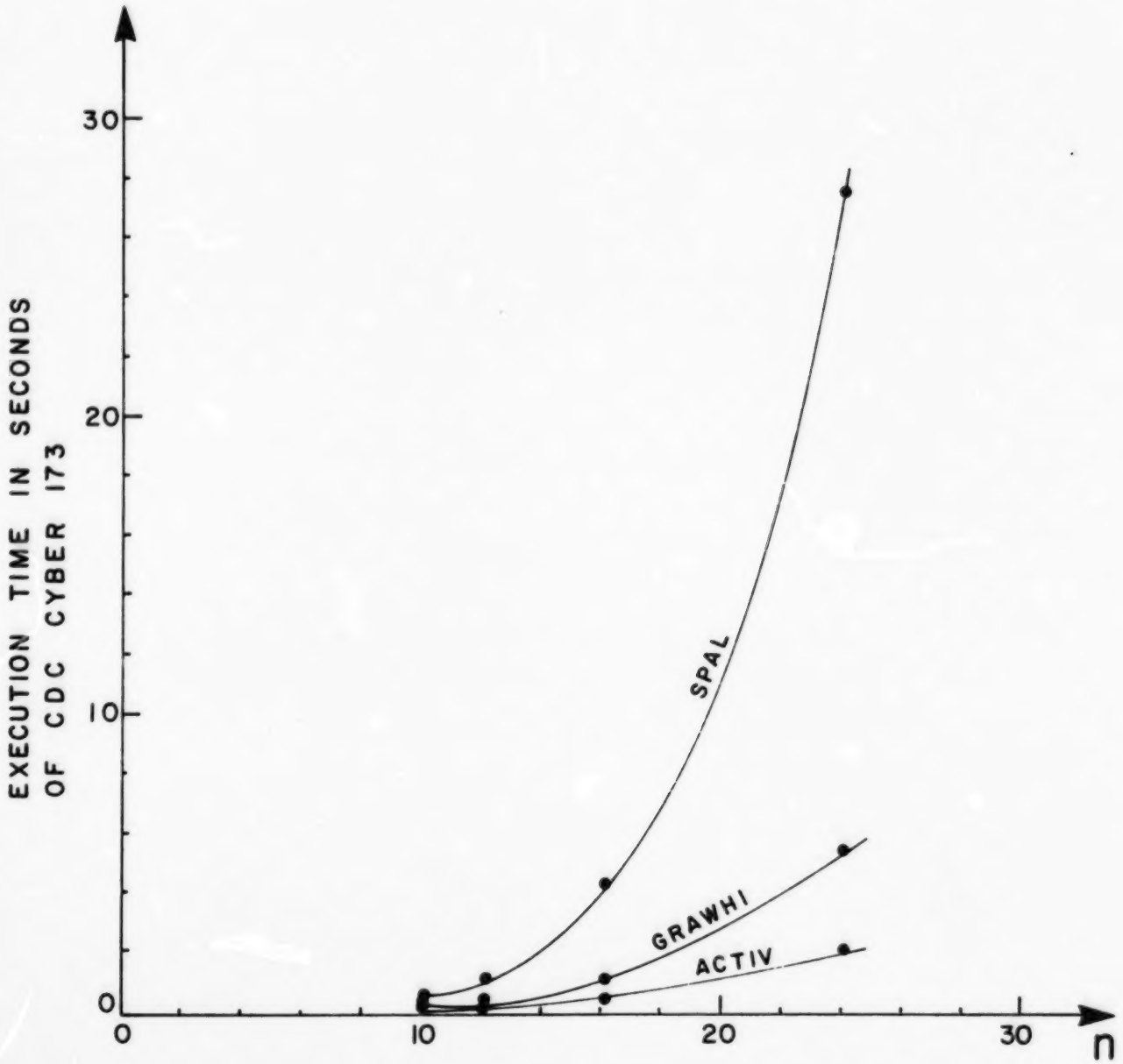


Figure 4: Comparison of the execution time of ACTIV, GRAWHI and SPAL

| Size n | Number of problems generated | Number of local optima generated per problem | Average number of exchanges during the algorithm | Average execution time in seconds of CDC Cyber 173 | Average percentage of improvement of the objective function |
|---|---|---|---|---|---|
| 10 | 13 | 12 | 4.54 | .125 | 14.18 |
| 12 | 20 | 12 | 6.05 | .215 | 13.40 |
| 24 | 5 | 24 | 18.4 | 2.08 | 11.34 |
| 36 | 5 | 36 | 34.0 | 8.05 | 10.09 |
| 48 | 5 | 5 | 48.0 | 19.61 | 9.00 |
| 54 | 2 | 10 | 58.0 | 29.66 | 8.66 |
| 57 | 1 | 1 | 55.0 | 31.74 | 8.65 |
| 58 | 1 | 1 | 61.0 | 35.87 | 7.54 |
| 60 | 1 | 1 | 51.0 | 33.20 | 7.90 |
| 72 | 1 | 1 | 103.0 | 89.32 | 8.32 |

Table 1: Results obtained by ACTIV for the first series of randomly generated problems of varying size n (Series B)

| | Size of the problem = n | Number of problems generated | Number of local optima generated per problem | Average number of exchanges during the algorithm | Average execution time in seconds of CDC Cyber 173 | Average percentage of improvement of the objective function |
|---|---|---|---|---|---|---|
| ACTIV | 10 | 20 | 12 | 4.55 | .121 | 14.03 |
| | 12 | 20 | 12 | 6.2 | .22 | 12.90 |
| | 16 | 5 | 16 | 10.2 | .56 | 12.26 |
| | 24 | 5 | 24 | 18.8 | 2.12 | 10.73 |
| SPAL | 10 | 20 | 12 | 3.9 | .54 | 14.39 |
| | 12 | 20 | 12 | 5.15 | 1.22 | 13.17 |
| | 16 | 5 | 16 | 7.8 | 4.35 | 12.64 |
| | 24 | 2 | 24 | 15.0 | 27.70 | 11.44 |
| GRAWH1 | 10 | 20 | | | .23 | |
| | 12 | 20 | | | .44 | |
| | 16 | 5 | | | 1.22 | |
| | 24 | 5 | | | 5.45 | |

Table 2: Results obtained by ACTIV, SPAL and GRAWH1 (Series C)[1]

1. For the problem of size n = 24, SPAL was applied to only 2 of the 5 problems generated

| Problem number | 1,3,5,6,7, 8, 9, 11, 12, 15, 16, 17, 19, 20 | 2 | 4 | 10 | 13 | 14 | 18 | Average percentage deviation from the optimum |
|---|---|---|---|---|---|---|---|---|
| ACTIV | 0.0 | 0.0 | .289 | .244 | .455 | .168 | .08 | .062 |
| SPAL | 0.0 | .070 | .047 | 0.0 | 0.0 | 0.0 | 0.0 | .006 |

Table 3: Percentage deviation between the global optimum and the best local optimum obtained by ACTIV or SPAL, for each of the 20 problems of size n = 10 (series C)

| n = 16 / Pb # | 1 | 2 | 3 | 4 | 5 | Average percentage of improvement of SPAL over ACTIV |
|---|---|---|---|---|---|---|
| Percentage of improvement of SPAL over ACTIV | 0.00 | 0.00 | .24 | .65 | .49 | .28 |

| n = 24 / Pb # | 1 | 2 | Average percentage of improvement of SPAL over ACTIV |
|---|---|---|---|
| Percentage of improvement of SPAL over ACTIV | .04 | .04 | .04 |

Table 4: Comparison of the quality of the best solutions produced by ACTIV and SPAL for the 5 problems of size n=16 (Series C) and for the 2 problems of size n=24 (Series C).

| Problem size n | ACTIV | | SPAL | |
|---|---|---|---|---|
| | Execution time in seconds of IBM 370/168 | Discrepancy between heuristic solution and optimum | Execution time in seconds of IBM 370/168 | Discrepancy between heuristic solution and optimum |
| 5 | 0.015 | none | | |
| 6 | 0.016 | none | | |
| 7 | 0.019 | none | | |
| 8 | 0.020 | 0.7% | | |
| 10 | 0.028 | 1.6% | | |
| 12 | 0.039 | 3.6% | 0.439 | 3.2% |
| 16 | 0.070 | Unknown | .727 | (Unknown but .77% better than ACTIV) |

Table 5: Comparison between SPAL and ACTIV on the problems of series A (only one local optimum was generated for each problem size)

| Permutation | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Average percentage deviation from the optimum |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ACTIV | 2.9 | 3.5 | 3.7 | 5.6 | 0.7 | 4.3 | 4.6 | 4.5 | 4.5 | 8.0 | 4.23% |
| SPAL | 3.2 | 3.5 | 4.0 | 3.1 | 7.9 | 3.8 | 2.2 | 2.1 | 2.8 | 2.3 | 3.49% |

Table 6: Comparison of the local optima produced by ACTIV and SPAL with the same 10 initial permutations on the problem of size n=12 (Series A).

| Method | ACTIV | SPAL | ACTIV + SPAL [1] |
|---|---|---|---|
| Cost | 133660788 | 133605798 | 133295008 |
| Execution time in seconds of CDC Cyber 173 | 3.06 | 35.50 | 7.64 |
| Number of exchanges implemented | 30 | 20 | 3 |

1. For ACTIV + SPAL, the execution time and the number of exchanges refer to SPAL only.

Table 7: Comparison of three methods for problem # 2 of size n=24 (Series C).

| Value of the initial assignment generated | Value of the local optimum | Execution time in seconds on CDC CYBER 74 | Number of exchanges implemented |
|---|---|---|---|
| 10212.67 | 4211.40 | 1.833 | 39 |
| 9344.56 | 4629.18 | 1.340 | 31 |
| 9850.57 | 4555.83 | 1.623 | 39 |
| 9963.64 | 4507.48 | 1.176 | 27 |
| 9419.66 | 4469.93 | 1.275 | 29 |
| 9234.90 | 4538.39 | 1.574 | 37 |
| 10331.25 | 4337.89 | 1.456 | 34 |
| 10116.79 | 4243.82 | 1.400 | 34 |
| 8575.05 | 4374.27 | 1.535 | 37 |
| 7790.89 | 4540.54 | 1.694 | 42 |
| 10560.13 | 4253.33 | 1.675 | 41 |
| 9069.98 | 4663.46 | 1.755 | 42 |
| 9830.87 | 4523.78 | 1.236 | 31 |
| 9014.68 | 4384.78 | 1.218 | 28 |
| 9151.23 | 4587.50 | 1.420 | 33 |

Table 8: Results of ACTIV on the Steinberg problem with 15 initial solutions chosen at random

## ACKNOWLEDGEMENTS

The author is grateful to Britton Harris for his help in the development of this work, and to François Shaigetz for his help with some of the more recent computer experiments.